Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 247 – 256

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Analysis of Users' Sentiments from Kannada Web Documents

K. M. Anil Kumar, N. Rajasimha*, Manovikas Reddy, A. Rajanarayana and Kewal Nadgir

Department of Computer Science, Sri Jayachamarajendra College of Engineering, Mysore 570 006, India

Abstract

In today's world, there is an explosive growth of data from terabytes to petabytes in the internet. The major problem is not the availability of data but starving for knowledge from the data. Sentiment analysis is an important current research area in the field of web content mining. Sentiment analysis and opinion mining is the study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language. In this paper, we extend our ideas pertaining to Sentiment Analysis to the regional language Kannada, spoken mainly in Karnataka, a state in southern part of India. We have explored the usefulness of semantic approaches and machine learning approaches, used predominately on English language data set, from Kannada web documents. We found the average accuracy of machine learning approaches to be better than the average accuracy of semantic learning approaches for Kannada data set.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Opinion; Review; Sentiment; Polarity; Negator; POS tagger; Translator.

1. Introduction

According to the dictionary, sentiment is defined as "an attitude, thought, or judgment prompted by feeling". Sentiment Analysis is a Natural Language Processing and Information Extraction task that aims to obtain writer's feelings expressed in positive or negative reviews, questions and requests, by analyzing a large numbers of documents. In a general sense, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall polarity of a document. The attitude may be his or her judgment or evaluation, affective state, or the intended emotional communication. In recent years, the exponential increase in the internet usage and exchange of public opinion is the driving force behind the need for Sentiment Analysis.

The analysis of sentiments may be document based where the sentiment in the entire document is summarized as positive, negative or objective. It can be sentence based where each and every sentence, having sentiments, in the text is classified. Sentiment Analysis can be phrase based where the phrases in a sentence are classified according to the polarity based on some patterns of their occurrence. Sentiments are classified as objective (facts), positive (denotes a state of happiness, bliss or satisfaction on part of the writer) or negative (denotes a state of sorrow, dejection or disappointment on part of the writer).

*Corresponding author. Tel.: 9739553772.
E-mail address: n.rajasimha392@gmail.com

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.029

We can observe the impact of social media and e-commerce in modern times, particularly with the advent of mobile phones. Accesses to social media and e-commerce sites have made access to opinions of others. Many use the opinions of others when buying or wanting to watch a movie or wanting to read a book. It must be noted that there are certain approaches for the classification of sentiments in English language. Also, there are no studies undertaken to find feasibility of existing approaches or develop new approaches for Indian languages. With popularity of sharing opinions in native languages across web sites, there is a need for Sentiment classification in native languages. For example, a native user may wish to write " ಇದು ತುಂಬಾ ಚೆನ್ನಾಗಿದೆ " instead of "This is very good" so that it benefits those who don't know English and still get the opinion of the reviewer. This paper aims in applying certain algorithms that are developed for sentiment analysis in English for a collection of Kannada opinions and also analyze the results. The paper is organized as follows: Section 2 discusses the related work, Section 3 explains the methodology, Section 4 is regarding experimental setup, Section 5 gives an account of the results obtained and lastly, Section 6 provides the concluding remarks.

2. Related Work

The analysis of data to extract latent public opinion and sentiment is a challenging task. Liu *et al.* (2009)[1] defines a sentiment or opinion as a quintuple

"<oj, fjk, soijkl, hi, tl>, where oj is a target object, fjk is a feature of the object oj, soijkl is the sentiment value of the opinion of the opinion holder hi on feature fjk of object oj at time tl, soijkl is +ve, -ve, or neutral, or a more granular rating, hi is an opinion holder, tl is the time when the opinion is expressed."

Pang-Lee *et al.* (2002)[2] broadly classifies the applications into the following categories.

- Applications to Review-Related Websites, Movie Reviews, Product Reviews etc.
- Applications as a Sub-Component Technology Detecting antagonistic, spam detection, context sensitive information detection etc.
- Applications in Business and Government Intelligence, Knowing Consumer attitudes and trends
- Applications across Different Domains Knowing public opinions.

According to Collomb *et al.*[3], the existing work on sentiment analysis can be classified from different points of views: technique used, view of the text, level of detail of text analysis, rating level, etc. From a technical point of view, we identify machine learning, lexicon-based, statistical and rule-based approaches.

- The lexicon-based approach involves calculating sentiment polarity for a review using the semantic orientation of words or sentences in the review.
- The rule-based approach looks for opinion words in a text and then classifies it based on the number of positive and negative words.
- The machine learning method uses several learning algorithms to determine the sentiment by training on a known dataset.
- Statistical models represent each review as a mixture of latent aspects and ratings. It is assumed that aspects and their ratings can be represented by multinomial distributions.

2.1 Kannada language transliteration

The Language transliteration is one of the most important area in natural language processing. Machine Transliteration is the conversion of a character or word from one language to another without losing its phonological characteristics. Kannada uses the UTF-8/western windows encode and draws its vocabulary mainly from Sanskrit language.

According to Mallamma V. Reddy[4], Kannada is a morphologically rich language in which morphemes combine with the root words in the form of suffixes. Kannada grammarians divide the words of the language into three categories namely:

Declinable words: Morphology of declinable words, as seen in many Dravidian languages is fairly simple compared to verbs. Kannada words are of three genders and also declinable and conjugable words have two numbers-singular and plural.

Conjugable words: The verb is much more complex than the nouns. There are three persons namely first, second and third person. Tense of verbs is past, present or future. Aspect may be simple, continuous or perfect.

Uninflected words: Uninflected words may be classified as adverbs, postpositions, conjunctions and interjections.

In Kannada, adjacent words are often joined and pronounced as one word which is called *Morphophonemics* or ಸಂಧಿ. *Composition* or ಸಮಾಸ is the process where two or more words combine together to form an overall new word which preserves the meaning of the combination of the words.

2.2 Part-of-speech tagging in Kannada

According to Vijayalaxmi F. Patil[5], in most of the Dravidian languages, particularly for Kannada language, nouns and verbs get inflected. Also verbs and adjectives are nominalized by means of certain nominalizers. Adjectives and adverbs do not inflect. So, many times we need to depend on syntactic function or context to decide upon whether the particular word is a noun or adjective or adverb or post position. This leads to the complexity in Kannada POS tagging. A noun may be categorized as common, proper or compound. Similarly, verb may be finite, infinite, gerund or contingent. Contingent form of verb is not found in other Dravidian languages except Kannada.

Siva Reddy and Serge Sharoff[6] have built a Hidden-Markov Model (HMM) based Kannada POS tagger. They use TnT, a popular implementation of the second-order Markov model for POS tagging and construct the TnT model by estimating transition and emission probabilities of Kannada using the cross-language Telugu. Their tagset has both POS and morphological information encoded in it, the HMM model has an advantage of using morphological information to predict the main POS tag, and the inverse, where main POS tag helps to predict the morphological information.

2.3 Tagset for Kannada

Vijayalaxmi F. Patil of LDC-IL proposed a Kannada tagset which consists of 39 tags with the following features: For each word, the grammatical categories as well as grammatical features are considered. Hence it needs to be split for each and every inflected word in the corpus. The number of tags is very large. This leads to increased complexity during POS tagging which in turn reduces the tagging accuracy. For simple POS level, a tagset which has just the grammatical categories excluding grammatical features and minimum tags without compromising on tagging efficiency. The proposed tagset as shown in Fig. 1 consists of tags where inflections are not considered. The compound tags are used only for nouns (NNC) and proper nouns (NNPC). There are 5 tags for nouns, 1 tag for pronoun, 8 tags for verbs, 3 for punctuations, two for number, and 1 for each adjective, adverb, conjunction, echo, reduplication, intensifier, postposition, emphasize, determiners, complimentizer and question word.

Akshar Bharati *et al.*[7] arrive at standard tagging scheme for POS tagging and chunking for annotating Indian languages (AnnCorra) and come up with the tags which are exhaustive for the task of annotation for Indian languages. They give a detailed description of the tags which have been defined for the tagging schemes and elaborate the motivations behind the selection of these tags. They have come up with tag names which are assigned by an existing tagger may be familiar to the users and thus are easier to adopt for a new language rather than a totally new one. The Penn tags are most commonly used tags for English. However, new tags have been introduced wherever Penn tags have been found inadequate for Indian language descriptions. For example, for verbs none of the Penn tags have been used. Instead, AnnCorra has only two tags for annotating verbs, VM (main verb) and VAUX (auxiliary verb)

2.4 Semantic techniques in sentiment analysis

2.4.1 Polarity classification

Subhabrata Mukherjee in his literature survey[8] mentions various techniques. A typical approach to sentiment analysis is to start with a lexicon of positive and negative words and phrases. In these lexicons, entries are tagged with their known prior polarity: out of context, does the word seem to evoke something positive or something negative. For example, ಸುಂದರ (beautiful) has a positive prior polarity, and ಘೋರ (horrid) has a negative prior polarity.

Sl No.	Category	Tag name	Example
1.1	Noun	NN	
1.2	NLoc	NST	
2.	Proper Noun	NNP	
3.1	Pronoun	PRP	
3.2	Demonstrative	DEM	
4	Verb-finite	VM	
5	Verb Aux	VAUX	
6	Adjective	JJ	
7	Adverb	RB	*Only manner adverb
8	Post position	PSP	
9	Particles	RP	bhI, to, hI, jI, hA.N, na,
10	Conjuncts	CC	bole (Bangla)
11	Question Words	WQ	
12.1	Quantifiers	QF	bahut, tho.DA, kam (Hindi)
12.2	Cardinal	QC	
12.3	Ordinal	QO	
12.4	Classifier	CL	
13	Intensifier	INTF	
14	Interjection	INJ	
15	Negation	NEG	
16	Quotative	UT	ani (Telugu), endru (Tamil), bole/mAne (Bangla), mhaNaje (Marathi), mAne (Hindi)
17	Sym	SYM	
18	Compounds	*C	
19	Reduplicative	RDP	
20	Echo	ECH	
21	Unknown	UNK	

Fig. 1. POS tag set for Indian languages (Nov. 2006, IIIT Hyderabad).

2.4.2 Negation

Negation can be expressed in subtle ways without the explicit use of any negative word. A method often followed in handling negation explicitly in sentences like “I do not like the movie”, is to reverse the polarity of all the words appearing after the negation operator (like not).

2.4.3 Adjectives only

Adjectives have been used most frequently as features amongst all parts of speech. A strong correlation between adjectives and subjectivity has been found. Although all the parts of speech are important people most commonly used adjectives to depict most of the sentiments and a high accuracy have been reported by all the works concentrating on only adjectives for feature generation.

2.4.4 Turney’s method

Turney *et al.*[9] present a simple unsupervised learning algorithm for classifying a review. The algorithm takes a written review as input and produces a speech classification as output. The first step is to use a part-of-tagger to identify phrases in the input text that contain adjectives or adverbs. The second step is to estimate the semantic orientation of each extracted phrase. The third step is to assign the given review to a class, recommended or not recommended, based on the average semantic orientation of the phrases extracted from the review. If the average is positive, the prediction is that the review recommends the item it discusses.

Two consecutive words are extracted from the review if their tags conform to any of the patterns in Fig. 2. The JJ tags indicate adjectives, the NN tags are nouns, the RB tags are adverbs, and the VB tags are verbs. The second pattern, for example, means that two consecutive words are extracted if the first word is an adverb and the second word is an adjective, but the third word (which is not extracted) cannot be a noun. NNP and NNPS (singular and plural proper nouns) are avoided, so that the names of the items in the review cannot influence the classification.

The second step is to estimate the semantic orientation of the extracted phrases, using the PMI-IR algorithm. This algorithm uses mutual information as a measure of the strength of semantic association between two words.

2.4.5 Sentence based approach

Khan and Baharudin[10] discuss about sentiment analysis at individual sentence level in which from subjective sentences, the opinion expressions are extracted and their semantic scores are checked using the SentiWordNet directory. The final weight of each individual sentence is calculated after considering the whole sentence structure.

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Fig. 2. Phrase patterns used for extracting value phrases – turney (2002).

2.5 Machine learning techniques in sentiment analysis

Jagtap and Pawar[11] explain that the machine learning approach involves text classification techniques. This approach treats the sentiment classification problem as a topic-based text classification problem. Any text classification algorithm can be employed, e.g., naïve Bayes, SVM, etc. This approach was put forth by Pang-Lee *et al.* (2002) to classify movie reviews into two classes: positive and negative. The Machine Learning algorithms that are considered for our experiments are:

J48 (C4.5) – It builds decision trees from a set of training data using the concept of information entropy. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other.

Random tree – It is a collection of tree predictors that is called forest further in this section. The random trees classifier takes the input feature vector, classifies it with every tree in the forest, and outputs the class label that received the majority of "votes".

ADT Tree – An alternating decision tree combines the simplicity of a single decision tree with the effectiveness of boosting. The knowledge representation combines tree stumps, a common model deployed in boosting, into a decision tree type structure.

Breadth First – Breadth First Search (BFS) searches breadth-wise in the problem space. Breadth first search expands nodes from the root of the tree and then generates one level of the tree at a time until a solution is found.

Naïve Bayes – Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. All naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

SVM – Support Vector Machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

Weka[12] is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka suite contains the Visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes. Weka is particularly used in machine learning as number of algorithms have been developed and implemented to extract information and discover knowledge patterns that may be useful for decision support.

3. Methodology

In the present scenario there doesn't exist formidable methods for review classification in Indian Languages like Kannada. For this purpose we adopt and develop certain algorithms for Kannada language which can be applied to analyze the sentiment expressed in a Kannada website. First, for the semantic methods, an exhaustive keyword list, both positive and negative ones separately is developed for which we follow manual identification in the review dataset and also the translation of English keywords using translator software like Google Translate into Kannada and also build a list of negators which is used in window algorithm. We make use of Kannada POS Tagger software for

```
Input : The set of positive/negative comments as separate files in a directory.
Output : Classified comments based on their polarity and total count of all 3 categories of classified comments.
  For each file in the directory
    Initialize temporary positive and negative counters of the comment in the file to zero
  For each line in the comment
      Tokenize the comment
      Initialize the counters to zero
    End For
  For each word in the file
   If the word is found in the Positive Keyword list
        Increment the positive counter by 1
  Else If the word is found in the Negative Keyword list
        Increment the negative counter by 1
End If
    End For
  Compute the polarity of the comment
EndFor
```

Algorithm 1. Baseline algorithm

```
For each word in the file
    If the word is found in the Positive/NegativeKeyword list
For each word within the window
If a negator is found then Decrement the positive/negative counter by 1
ElseIncrement the positive/negative counter by 1
    End If
End For
    End If
End For
```

Algorithm 2. Negator-window algorithm

```
Tag the comment using POS Tagger
For each word in the file
If the word is tagged as JJthenCompute the polarity
   End If
End For
```

Algorithm 3. POS tagging algorithm

implementing adjective analysis and Turney's algorithm. Stanford POS Tagger are employed for tagging the translated reviews required for applying the adjective analysis and Turney's algorithm for the corresponding translated English review of each Kannada review in the dataset. Lastly, sentence level approaches are experimented on the dataset by splitting the review into individual sentences. Then, we try out a few machine learning algorithms like J48, Random Tree, ADT Tree, Breadth First, Naïve Bayes and Support Vector Machine in the Weka software and compare the obtained results with the semantic methods.

In the baseline algorithm, the polarity is computed by taking the difference between positive and negative counts. If the value results to be more than zero, then the comment is classified as positive, if it is less than zero then it is classified as negative otherwise as neutral.

As an enhancement to the above method, algorithm searches a negator for each encountered keyword within a window size i.e. some fixed number of words before and after the keyword. The negators are stored in a separate file. In the below segment of the algorithm, we show the actions taken.

The next method which we adopt is the POS Tagging method. Here, we first tag the comment using a POS Tagger software tool[6] and then analyze only those words which are tagged as adjectives, by applying the baseline method.

Further to the POS method, we try to recognise various patterns formed by the words (Refer Fig. 2) as developed by Turney and when a pattern is found, we try to find keyword in the pattern and find out the polarity.

Tag the comment using POS Tagger
For each phrase in the comment
If the phrase matches any of the Turney's pattern **then**Compute the polarity by analysing the phrase
End If
End For

Algorithm 4. Turney's algorithm

Split the comment into separate sentences
For each sentence of the comment
Compute the number of keywords (both positive and negative)
End For
Find the sentence with maximum number of keywords
Compute the polarity for the sentence
If there is a tie in the maximum number
Compute the average of the polarities
End If

Algorithm 5. Significant sentence algorithm

We also try to apply the above two algorithms for the translated reviews where we first feed the reviews to a standard translator software like Google Translate and obtain the translated reviews in English. We then tag the translated reviews using English POS Tagger tool like Stanford POS Tagger and for a standard collection of English keywords[13].

Finally, we experiment the sentence based approach, where we find out the keyword count and polarity for each sentence in the review. The sentence having the maximum number of keywords is considered as the most significant sentence and therefore its polarity will be projected as the polarity of the whole review. If a tie occurs, then average is taken. As special cases of the algorithm, we consider three different situations, where we consider only the first sentence, only the last sentence and the middle sentence(s) as the significant sentences assuming that they have the highest number of keywords and therefore have a larger sentimental effect on the overall comment.

The machine learning methods are applied using Weka software which is discussed earlier.

4. Experimental Setup

In this paper, as a part of the experiment, we collected 182 positive Kannada reviews and 105 negative Kannada reviews as our text corpus and tried out all the mentioned algorithms and analyzed the results. For the Negator-Window Algorithm, we took window sizes of 3, 5 and 7 and for the Sentence Analysis Algorithm, we considered three special cases by taking the first, middle and last sentence as significant sentences and tried the baseline algorithm for the special cases along with the actual most significant sentence. The reviews were mainly collected for broad domains consisting of commercial products like automobiles, health and body care products like soaps, shampoo, electronic items like TV, mobiles, movies, songs, websites, TV programs, famous people, etc. We collected the keywords both manually as well as translating the available English keyword set in the website using Google Translate. In a similar way, we have collected a few negators which reverse the polarities of the keywords such as ಅಲ್ಲ (no), ಅಲ್ಲ (not), ಸುಳ್ಳು (false) with inputs from 10 evaluators and with an agreement of 60%.

All the Kannada reviews were stored as separate files in two separate folders one each for positive set and negative set. The programs were designed to take review input as files from the folders. All the Kannada reviews were entered in UTF-8 format. The Kannada POS tagger was used to tag the words of the files with their part-of-speech. The reviews were translated into English using Google Translate and we used Stanford POS tagger to tag the translated English reviews and applied methods like POS tagging and Turney patterns for the same.

We also tried to classify the reviews using Weka software (version 3.6.10) and the results are thus obtained using various machine learning algorithms like J-48, random Tree, ADT Tree, Breadth First, Naïve Bayes and Support vector machine (SMO in Weka) algorithms. This allows systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of data sets. Weka also allows us to set large scale experiments, start them running, leave them, and they analyze the performance statistics that have been collected and finally they automate the experimental process. We can use the RemovePercentage filter in the preprocess panel and split the whole dataset into

Table 1. Evaluation results of (a) Semantic approaches; (b) Machine learning approaches.

Algorithm	Positive Review Set			Negative Review Set			Overall		
	TP Rate	FP Rate	Precision	TP Rate	FP Rate	Precision	TP Rate	FP Rate	Precision
Baseline	0.96	0.21	0.888	0.79	0.04	0.912	0.898	0.148	0.897
Window 3	0.90	0.18	0.896	0.82	0.10	0.82	0.871	0.151	0.868
Window 5	0.86	0.26	0.853	0.74	0.14	0.757	0.816	0.216	0.818
Window 7	0.83	0.34	0.807	0.66	0.17	0.69	0.768	0.278	0.764
POS Kan.	0.66	0.75	0.605	0.25	0.34	0.299	0.510	0.60	0.493
POS Eng.	0.65	0.59	0.655	0.41	0.35	0.401	0.562	0.502	0.562
TurneyKan.	0.56	0.79	0.551	0.21	0.44	0.216	0.432	0.662	0.429
TurneyEng.	0.59	0.60	0.632	0.40	0.41	0.362	0.520	0.530	0.533
Sen_start	0.53	0.58	0.533	0.42	0.47	0.341	0.490	0.540	0.730
Sen_mid	0.66	0.35	0.766	0.65	0.34	0.527	0.656	0.346	0.679
Sen_end	0.59	0.45	0.697	0.55	0.41	0.439	0.575	0.435	0.603
Sen_sig	0.92	0.34	0.824	0.66	0.08	0.831	0.825	0.245	0.827

(a)

Algorithm	Positive Review Set			Negative Review Set			Overall		
	TP Rate	FP Rate	Precision	TP Rate	FP Rate	Precision	TP Rate	FP Rate	Precision
J48	0.76	0.22	0.656	0.78	0.24	0.858	0.767	0.227	0.729
Random Tree	0.83	0.54	0.607	0.46	0.17	0.726	0.695	0.405	0.650
ADT Tree	0.88	0.37	0.758	0.63	0.12	0.805	0.788	0.278	0.775
Breadth First	0.89	0.35	0.773	0.65	0.11	0.814	0.802	0.262	0.788
Naïve Bayes	0.83	0.23	0.863	0.77	0.17	0.723	0.808	0.207	0.812
SMO	0.85	0.33	0.816	0.67	0.15	0.722	0.784	0.266	0.781

(b)

training and testing dataset, which we used to split our dataset into two equal halves, one for training and the other for testing.

5. Evaluation Results

We have applied the above methods for the review classification in both positive and negative review sets and obtained the above results. Table 1(a) gives the comparative results for semantic methods and Table 1(b) similarly gives the comparative results for machine learning methods. The parameters used to evaluate and compare our methods are:

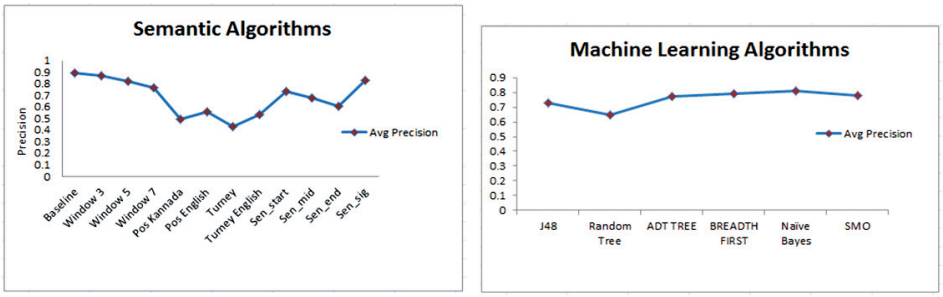
TP Rate, FP Rate and Precision. Firstly we obtain the confusion matrix that allows visualization of the performance of an algorithm. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

The *True Positive* (TP) rate is the proportion of examples which were classified as class x, among all examples which truly have class x, i.e. how much part of the class was captured. In the confusion matrix, this is the diagonal element divided by the sum over the relevant row. The *False Positive* (FP) rate is the proportion of examples which were classified as class x, but belong to a different class, among all examples which are not of class x. In the matrix, this is the column sum of class x minus the diagonal element, divided by the rows sums of all other classes. The *Precision* is the proportion of the examples which truly have class x among all those which were classified as class x. In the matrix, this is the diagonal element divided by the sum over the relevant column. For example, The TP Rate calculated from the above definition is 174/(174+8), which is 0.96, the FP Rate is 174/(174+8), which is 0.96 and the precision is 174/(174+8), which is 0.96.Similarly, for the rest of the other methods also we thereby obtained the confusion matrix and calculated the parameters.

We have found out that the baseline method outperforms all the other approaches as we have a well-trained data set. When we consider the window algorithm applied to our data set we find out that window 3 is more accurate when compared to window 5 and window 7. Among the sentence based approaches significant sentence fares well compared to the other three sentence based approaches. We have also found out that POS English, Turney English Pattern methods are better than POS Kannada, Turney Kannada methods respectively. Also, we found that classification of positive reviews were more accurate than classification of negative reviews.

Among the machine learning methods, nearly all the algorithms performed well and Naïve Bayes gave the better result compared to others in Weka in terms of accuracy. As all the methods are supervised learning methods, we can see that the results are based on the training set which actually infers based on the learning algorithm.

It can be observed from Table 1(a) that the semantic approaches perform well in case of Baseline Algorithm (precision – 0.897 or 89.7%) and Negator-Window Algorithm (Average precision of all window sizes – 0.816 or



(a) (b)

Fig. 3. Average precision graph of (a) Semantic algorithms; (b) Machine learning algorithms.

81.6%). Also, the Sentence approach in case of significant sentence performs well (precision – 0.827 or 82.7%). The average precision of all semantic methods combined is 0.6836 (or 68.36%).

When it comes to machine learning approaches it is evident, from Table 1(b), that all methods perform well, though to top it all, Naïve Bayes (precision – 0.812 or 81.2%) has the best results. The average precision of all the machine learning put together is 0.7558 or 75.58%. The reason for the better results from machine learning methods is due to the fact that they are designed to learn from the training dataset and don't depend on explicit patterns.

We find that the average precision of methods under machine learning approach is 7.22% better than methods under semantic approach.

Figures 3(a) and (b) show the comparison of various algorithms used in for Sentiment Analysis of Kannada reviews. We have analyzed some special cases wherein a few semantic algorithms don't perform well in certain situations. For example, in negator-window algorithm, we may come across negator overlap case where a single negator will be part of two separate windows due to which output is affected. The ability of the tagger to split the words plays a very important role in determining the part of speech. For example, ಚೆನ್ನಾಗಿದೆ is as a single word tagged as verb, whereas when the word is split as ಚೆನ್ನಾಗಿ and ಇದೆ, the former is tagged as adverb and the latter is tagged as verb. Thus, this is a very important factor for the low performance of algorithms like Turney and POS adjective analysis since the ***morphophonemics*** of Kannada result in word compounds. Also, the correctness of translator plays an important role in the correct functioning of algorithms applied for translated reviews. All the above factors have an impact of the semantic approaches on their performance.

6. Conclusion and Future Work

We have presented a few methods under semantic and machine learning approaches for finding users sentiments. In semantic methods, the Baseline, Neagtor-Window and Sentence based methods perform well and in machine learning methods, Naïve Bayes method performs the best. These methods can be further enhanced to classify reviews still more accurately. It can be seen that these methods aid the commercial websites to classify Kannada reviews automatically without human intervention. It is designed to assist those people who are unable to read English opinions and can understand only native languages. The limitations that we faced in our methods are due to the unavailability of an exhaustive keyword list, a perfect POS tagger and an accurate translation tool. We can also see that certain methods (like Turney's approach) don't perform well since the phrase patterns have different structure in Indian languages and also context sensitive interpretation hinders the accuracy.

The present work can be further improved by developing a broader list of keywords and develop patterns which are more suitable for native languages. Any web application can incorporate these algorithms and the keyword database to enable the intelligent classification of opinion by the website which allows Kannada reviews to be entered. It is worthwhile to note that we can actually extend this work to other Indian languages like Telugu, Tamil, Hindi, Malayalam, etc. as well since the basic structure of sentences and words are similar among majority of Indian languages.

References

[1] Ramanathan Narayanan, Bing Liu and Alok Choudhary, Sentiment Analysis of Conditional Sentences, In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, (2009).

[2] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, (2008).

[3] Crina Costea, Damien Joyeux, Omar Hasan and Lionel Brunie, A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation by Anaïs Collomb.

[4] Mallamma V. Reddy and M. Hanumanthappa, Indic Language Machine Translation Tool for NLP.

[5] Vijayalaxmi F. Patil, Designing POS Tagset for Kannada, LDC-IL, CIIL Mysore.

[6] Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources by Siva Reddy and Serge Sharoff.

[7] Akshar Bharati, Dipti Misra Sharma, Lakshmi Bai and Rajeev Sangal, AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages, *Language Technologies Research Centre IIIT*, Hyderabad.

[8] Subhabrata Mukherjee, Sentiment Analysis: A Literature Survey, IIT Bombay.

[9] P. Turney, Thumbs up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, (2002).

[10] Aurangzeb Khan and Baharum Baharudin, Sentiment Classification by Sentence Level Semantic Orientation using SentiWordNet from Online Reviews and Blogs, Universiti Teknologi PETRONAS Perak, Malaysia.

[11] V. S. Jagtap and Karishma Pawar, Analysis of Different Approaches to Sentence-Level Sentiment Classification.

[12] http://www.cs.waikato.ac.nz/ml/weka/

[13] https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107/blob/master/data/opinion-lexicon-English/positive-words.txt
https://raw.github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107/master/data/opinion-lexicon-English/negative-words.txt